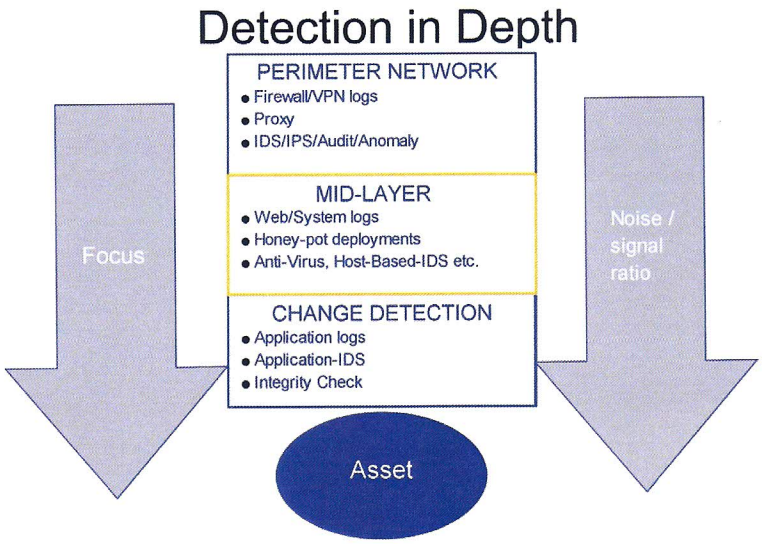
Learning Objective 15

LO15: Describe detection in depth – mid-layer



Web and System Logs

Operation systems and applications provide a wealth of logging information. This information can be used to monitor the health of the system and potentially detect malicious activity.

As the type, origin, and sophistication of attacks against computer networks has changed significantly, changes in techniques for auditing and logging for host, application, data store, and user access control requires significant upgrades as well to improve network monitoring to be able to detect advanced intrusion attempts.

If you can spend an hour, or 30 minutes at a minimum, to look at your logs on a daily basis, you will be in a better position to identify unusual entries. Once detected you can react. If you are not watching your logs regularly, you will not be able to catch potential problems.

Log Sources

Typical System logs: Routers, Antivirus, Syslog (**ix), Windows Event Log, NetFlow.

Typical Web Servers: IIS Logs, Apache Logs, Nginx, Oracle Web Server.

Windows System Logs

Each Windows Server maintains its own set of logs captured by Windows Event Viewer. The Application Log captures application information, warning, and error events. The System Log captures system-level information, warning, and error events. The Security Log captures audit events in the system including successful logins, unsuccessful logins, user-level privilege changes, and system-level

policy changes.

The same data can provide valuable IDS type of information to the security analyst.

Windows has a proprietary web server called Internet Information Services (IIS). IIS uses a flexible and efficient logging architecture. When a loggable event occurs, IIS writes to one of the logs stored in %SystemRoot%\system32\Logfiles\<service_name>.

IIS writes files in one of several formats:

- Centralized binary logging format
- W3C extended log file format (most commonly used)
- NCSA common log file format
- IIS log file format
- ODBC log file format

Linux System Logs (Syslog)

Linux and most other *nix variants use a logging system called syslog. Syslog was originally designed to use UDP because of its simplistic nature and low network overhead. Modern systems are turning to sending syslog formatted packets over TCP and in some cases using Transport Layer Security (TLS). Most network devices such as routers, firewalls, and IDSs provide log files over the syslog protocol.

When searching for a network log management product, the syslog should be an important data log format for consideration. Syslog allows organizations to stand up a log collection infrastructure without needing to coordinate the log transmission capabilities of a wide variety of log providers.

Any network log management product worth considering should have syslog as one of its log import mechanisms. Syslog allows organizations to stand up a log collection infrastructure without needing to coordinate the log transmission capabilities of a wide variety of log providers

Linux system logs (Syslog) are usually found in /var/log.

- messages: General message and system-related stuff
- auth.log: Authentication logs
- maillog: email logs
- access.log: Apache access log (found in /var/log/apache2)
- error.log: Apache error (found in /var/log/apache2)
- boot.log : System boot log
- mysqld.log: MySQL database server log file
- +others

Syslog was developed in the 1980s by Eric Allman as part of the Sendmail (first e-mail) project, and was initially used solely for the Sendmail. It proved so valuable that other applications began using it. In August of 2001, the Internet Engineering Task Force documented the status quo in RFC 3164. In March 2009, the original RFC was made obsolete by subsequent additions in RFC 5424.

The RFC can be found at: http://tools.ietf.org/html/rfc5424.

Log Correlation Can Help Locate Problems

Log correlations can help locate problems. This information can be used to monitor the health of the system and potentially detect malicious activity. It is important to:

- Examine server and AD logs to find login attempts. Look for unusual situations, such as the president logging in from a Starbucks in England, when he is actually in the middle of a safari in Africa.
- Look for SSH sessions over nonstandard (unusual) ports.
- Try to correlate complaints that "things" are slow, by monitoring help/operator desk calls.
- "Autopsy" systems/processes that stop working.

Examples of Logs and Indicators of Compromise

Below are some examples of web server logs that show indicators of compromise. The web server logs are shown in Apache 2 format (a common web server) and in Bro IDS format. Bro IDS has the capability to create logs from network traffic that is in clear text (not SSL).

NOTE: *Logs are typically on one line. For display purposes we have broken the log entry into multiple lines.*

Apache2 Access log format:

1	id.orig_host_ip	addr
2	Identity(Not Used)	string
3	UserName	string
4	Timestamp	time
5	Request	string
6	status_code	count
7	status_msg	string
8	Referer	string
9	User Agent	string

Bro format:

Bro log format: field #, Parameter, Type

1	Timestamp	time	15	status_code	count
2	uid	string	16	status_msg	string
3	id.orig_host_ip	addr	17	info_code	count
4	id.orig_port	port	18	info_msg	string
5	id.resp_host_ip	addr	19	filename	string
6	id.resp_port	port	20	tags	set[enum]
7	trans_depth	count	21	username	string
8	method	string	22	password	string
9	host	string	23	proxied	set[string]
10	uri	string	24	orig_fuids	vector[string]
11	referrer	string	25	orig_mime_types	vector[string]
12	user_agent	string	26	resp_fuids	vector[string]
13	request_body_len	count	27	resp_mime_types	vector[string]
14	response_body_len	count			

Example 1: cmd.exe sent over the network

Apache2 Server Log:

1.2.3.219 - - [07/May/2015:14:50:29 +0100] "GET

/scripts/..\xc0\xaf..\xc0\xaf..\xc0\xaf..\xc0\xaf..\xc0\xaf../winnt/system32/

cmd.exe?/c+dir+c:\+/OG" 404 4040 - - "Mozilla/4.0 (compatible; MSIE 8.0; Windows

NT 5.1; Trident/4.0)"

Bro IDS Log:

1431010229.218906 C3lqIb2PrXQkP6HIC4 1.2.3.219 40248 1.2.3.20 80 1 GET 1.2.3.20 /scripts/..\xc0\xaf..\xc0\xaf..\xc0\xaf..\xc0\xaf..\xc0\xaf../winnt/system32/ cmd.exe?/c+dir+c:\+/OG -Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0) 0 4040 404 Object Not Found - - - (empty) - - - - - FEvshPjWGgrdWzJna text/html

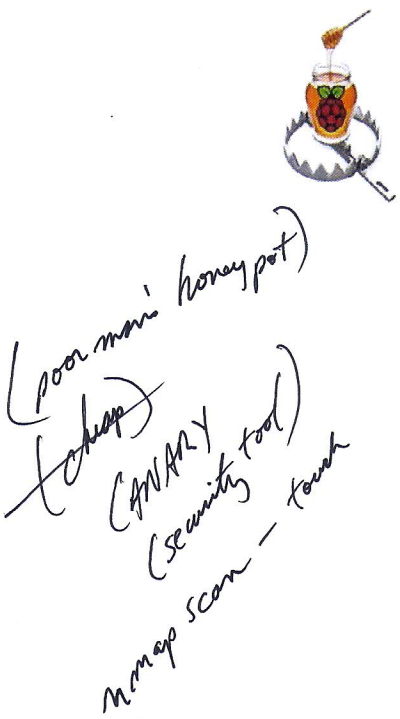
Example 2: SQL Injection

Apache2 Server Log:

2.3.5.89 - - [01/Jun/2015:11:29:21 -0600] "GET /forum/thread.php?search=something%22+UNION+Select+1%2C2%2Cpassword%2C4%2C5%2Cusername%2C7+from+users%3B%23&how=comment&thesearch=Search HTTP/1.1" 200 1375 - "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; .NET CLR 3.0.30729)"

Bro IDS Log:

1433179834.278614 CxjhFe22NyzbeXhEEi 2.3.5.89 59379 1.2.3.6 80 1 GET 1.2.2.90 /forum/thread.php?search=something"+UNION+Select+1,2,password,4,5,username,7+from+users;#&how=comment&thesearch=Search - Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; .NET CLR 3.0.30729) 0 5025 200 OK- - - HTTP::URI_SQLI - - - - - FnqS5i1NtFwUoa9oX2 text/html



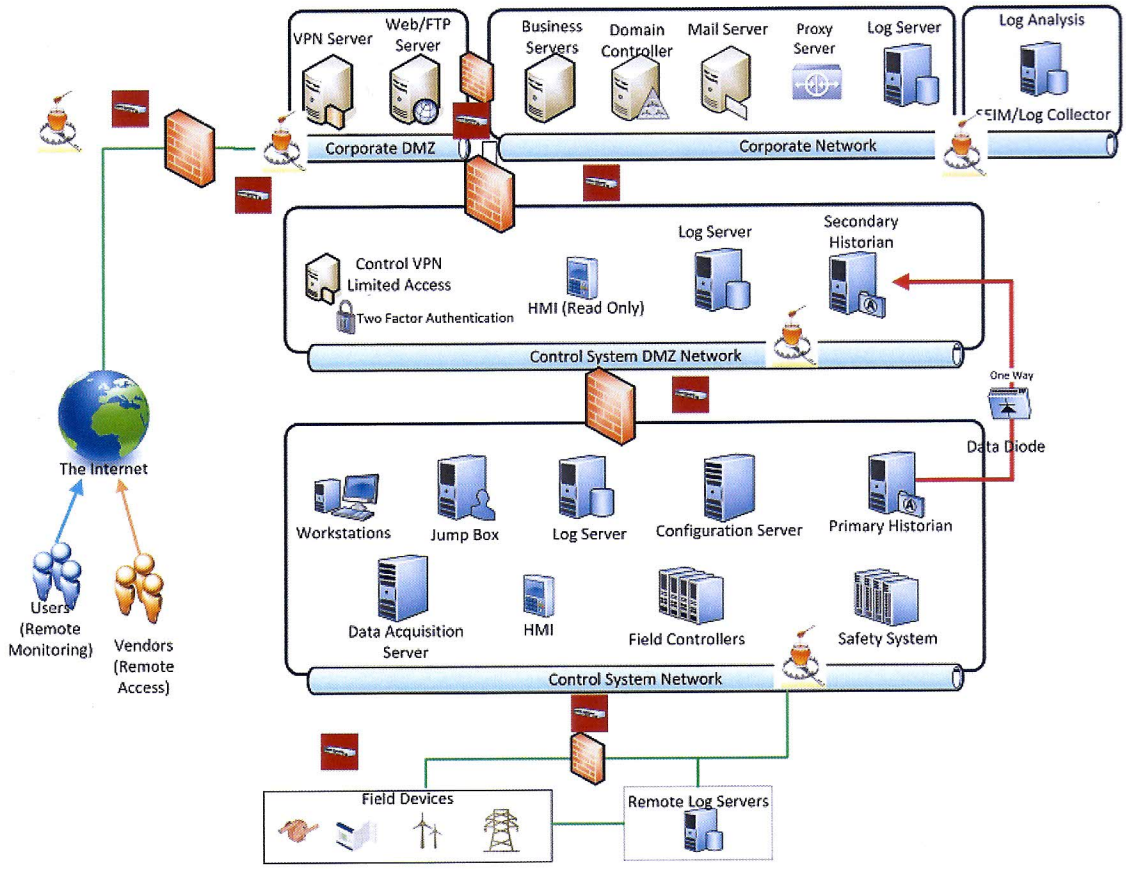
Honey Pot – A Decoy System

Honey Pots are decoy systems setup to make the 'enemy' think they have attacked a real system but instead they are hammering away at a fake. Honey Pots are a variant of an IDS, but with more of a focus on information gathering and deception. They exist as cybersecurity tool/device.

Often times you will hear of a very simple honey pot called a ***canary***. The idea behind a canary is that it doesn't communicate with any other system on your network. If an IDS is watching for ANY traffic to/from the canary, you will get an early warning that something is going on that shouldn't be. The question becomes, **who is doing it**, not **are they doing it**?

One easy way to install a canary is to add an additional NIC card to the system and assign it a never used IP address.

In the illustration on the next page, we have added honey pots to our suggested network diagram to illustrate their placement.



The most know world-wide honey pot project is the Honeynet Project, started in 1999. It is the world's largest and only distributed honey pot network.

One of the new projects of the project is Conpot, an ICS/SCADA Honeypot.

According to their webpage (http://conpot.org):

"Conpot is a low interactive server side Industrial Control Systems honeypot designed to be easy to deploy, modify and extend. By providing a range of common industrial control protocols we created the basics to build your own system, capable to emulate complex infrastructures to convince an adversary that he just found a huge industrial complex. To improve the deceptive capabilities, we also provided the possibility to server a custom human machine interface to increase the honeypots attack surface. The response times of the services can be artificially delayed to mimic the behavior of a system under constant load. Because we are providing complete stacks of the protocols, Conpot can be accessed with productive HMI's or extended with real hardware. "

The project is open source and available via github at: https://github.com/glastopf/conpot.

For more information go to their home page at http://www.honeynet.org.

Antivirus, Host Based IDS (HIDS)

Host based intrusion detection (HIDS) refers to intrusion detection that takes place on a single host system. HIDS involves installing an agent on the local host that monitors and reports on the system configuration and application activity. Some common abilities of HIDS systems include:

- Virus detection/mitigation
- Log analysis
- Event correlation
- File/system integrity checking
- Policy monitoring/enforcement
- Network monitoring from the host viewpoint
- Active response
- Rootkit detection
- Real-time alerting.

HIDS often have the ability to baseline a host system to detect variations in system configuration. In specific vendor implementations, these HIDS agents also allow connectivity to other security systems. This allows for central management of configuration policy and verification.

To be effective in an environment with more than a few hosts, HIDS are generally deployed to be managed from a central location. On the management system a policy is configured for deployment to local agents. There can be a single policy for all computers, but in most environments there will likely be multiple policies for particular operating systems, machine types, physical locations, and user types.

The central management of the systems is the key to success and/or failure. There must be a knowledgeable/competent active directory administrator to make this work.

Most modern HIDS packages have the ability to **actively** prevent malicious or anomalous activity on the host system. Due to the potential impact on the end user, HIDS tools are initially deployed in "monitor only" mode. This enables the administrator to create a baseline of the system configuration and activity. Active blocking of applications, system changes, and network activity is limited to only the most egregious activities. The policy can then be tuned based on what is considered "normal activity." Once a policy is configured, it is then applied and distributed to the hosts. Benefits of this central management architecture are:

- Ability to apply changes to many systems at once

- Create a "baseline" for known system types/use cases
- Central authentication, alerting, and reporting
- Central audit logging.

The main two issues with using any HIDS in an ICS environment are:

- Does my Operating System even support the use of a HIDS?
- Do I have enough hardware capacity to support the HIDS (cpu, memory, network bandwidth, etc.)

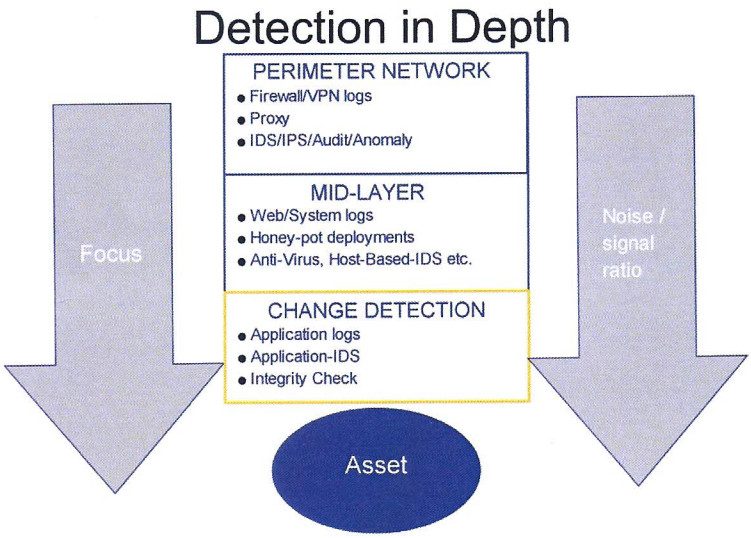
Bottom line: **Test any HIDS deployment in a sandbox, not your operational environment!**

See "Host Integrity Monitoring: Best Practices for Deployment" written in 2004, found at: http://www.symantec.com/connect/articles/host-integrity-monitoring-best-practices-deployment

Even though it is old, the concepts are the same.

Learning Objective 16

LO16: Describe the components of change detection.



Application Logs

Most applications will generate logs that give you information on how it is running, accesses, even status. This information can be used to determine if a change has been made that could adversely affect your security posture. Using this information in conjunction with network and system logs you can identify a compromise and attacker. For example, databases typically hold the most important pieces of an organization. Unfortunately, database security is often lacking, leaving sensitive information such as customer data, control system data and more, vulnerable to hackers.

Databases are now becoming one of the most voluminous log generators in the enterprise – rivaling firewalls for the top spot. Database logging thereby becomes an essential (and required) component of database security – and it makes sense to not only focus on keeping the bad guys out, but also to take a "what's going

on in here?" approach. After all, you may not know who the bad guys are. Logs will point you to the who, what, when, and where information of any breach – whether the malicious behavior comes from outside hackers, a disgruntled employee, or another source.

Typical database log events may include:

- User logins and logouts
- Database system starts, stops and restarts
- Various system failures and errors
- User privilege changes
- Database structure changes
- Most other DBA actions
- Select or all database data access (if configured to be so).

As we know, hackers are always looking for new ways to break through security barriers to access your sensitive information, and all preventative security measures fail at some point. Thus, because you are not able to guard against every malicious hacker, logs will at least allow you to detect such security breaches as well as actually figure out how it was done during the incident investigation.

Regularly collecting log data is a best practice for incident response and can save you during crunch time after a server crash, data theft, or surprise visit by your friendly auditor. Alternatively, if someone is downloading an entire table or changing a database schema while being logged on from a remote connection, a real-time alert will catch your attention. Further, reports may help you track and analyze login failures and successes or after-hours access to better evaluate insider privilege abuse. In other words, database logs can help you catch unusual behavior before a problem gets out of hand and into headline news.

Database log management is becoming a best practice for database security – you should be aware of who is accessing or changing your data, when they are accessing it, and where they are accessing it. When you combine your application logs with network, system and other logs you will have a complete picture of what has happened during an incident.

Application IDS/IPS/Firewall

While a host IDS/IPS would normally concentrate on protecting the host's operating system, as the name suggests, an application IDS/IPS will work solely with the application itself. They tend to be tailored to a specific product, such as, Microsoft Internet Information Server (IIS) within application groups that provide externally visible services such as Webservers, databases, and

mailservers. An IDS will report when nefarious activity is detected most usually using logs generated by the application, whilst an application IPS will not only detect such activity but also block it, protecting the application from attack.

There are still some drawbacks to an IPS. IPSs are designed to block certain types of traffic that it can identify as potentially bad traffic. IPSs **DO NOT** have the ability to understand web application protocol logic. At the application layer (OSI Layer 7), IPSs cannot fully distinguish if a request is normal or malformed. This could potentially allow attacks through without detection or prevention; especially newer attacks where signatures are not available (think zero-day).

Web Application Firewall

One example of an application IDS/IPS is a Web Application Firewalls (WAFs). WAFs are designed to protect web applications/servers from web-based attacks that a network IPS cannot prevent. WAFs are typically deployed in some sort of proxy fashion just in front of the web applications, so they do not see all traffic on our networks. WAFs are a special breed that can used to detect/prevent attacks against web applications in more depth than an IPS.

Unlike IPSs, WAFs interrogate the behavior and logic of what is requested and returned. WAFs protect against web application threats like SQL injection, cross-site scripting, session hijacking, parameter or URL tampering, and buffer overflows.

Some Web Applicantion Firewalls include:

AQTRONIX WebKnight - AQTRONIX WebKnight is an application firewall for IIS and other web servers and is released under the GNU General Public License. More particularly it is an ISAPI filter that secures your web server by blocking certain requests. If an alert is triggered WebKnight will take over and protect the web server.

https://www.aqtronix.com/?PageID=99

ESAPI - ESAPI (The OWASP Enterprise Security API) is a free, open source, web application security control library that makes it easier for programmers to write lower-risk applications. The ESAPI libraries are designed to make it easier for programmers to retrofit security into existing applications. The ESAPI libraries also serve as a solid foundation for new development.

https://www.owasp.org/index.php/Category:OWASP_Enterprise_Security_API#tab=Home

IronBee - IronBee is a universal web application security framework. Think of IronBee as a framework for developing a system for securing web applications - a framework for building a web application firewall (WAF) if you will.

http://www.ironbee.com

http://guardian.jumperz.net/index.html

Jumperz_net - JUMPERZ_NET is an open source application layer firewall for HTTP/HTTPS. It works as a reverse proxy server. It analyzes all HTTP/HTTPS traffic against rule-based signatures and protects web servers and web applications from attack. When unauthorized activity is detected, JUMPERZ_NET can disconnect the TCP connection before the malicious request reaches the web server.

http://www.modsecurity.org/

ModSecurity - ModSecurity is a cross-platform toolkit for real-time web application monitoring, logging, and access control. Think of it as an enabler: there are no hard rules telling you what to do; instead, it is up to you to choose your own path through the available features.

The following is a short list of the most important usage scenarios:

- Real-time application security monitoring and access control
- Virtual patching
- Full HTTP traffic logging
- Continuous passive security assessment
- Web application hardening
- Something small, yet very important to you.

https://www.owasp.org/index.php/Category:OWASP_ModSecurity_Core_Rule_Set_Project

The OWASP ModSecurity CRS Project's goal is to provide an easily "pluggable" set of generic attack detection rules that provide a base level of protection for any web application. The project is a set of web application defense rules for the open source ***ModSecurity***.

http://www.smoothwall.org/about/

Smoothwall - The Smoothwall Open Source Project was set up in 2000 to develop and maintain Smoothwall Express - a Free firewall that includes its own security-hardened GNU/Linux operating system and an easy-to-use web interface.

http://mvnrepository.com/artifact/org.webcastellum/webcastellum/1.8.3

WebCastellum - Java-based Open Source WAF (Web Application Firewall) to include inside a web application in order to protect it against attacks like SQL Injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), Parameter Manipulation and many more.

Integrity Checking

Integrity checking is a method of change detection where the current state of stored data is compared to a previously known state. Generally the *state* information that is used for comparison is a form of a hash. The reasoning behind this is if something bad happens on a system then something must have changed. To detect when something bad happens simply requires detecting changes

that shouldn't happened.

Integrity checking is generally considered one of the strongest anti-malware controls, since it has the potential to detect and locate all persistent malware intrusions along with any additional persistent changes to the system that malware could have made.

File verification is the process of verifying that a file has not been changed and/or that two files are identical, bit by bit. Both ***md5sum*** and ***sha1sum*** create "fingerprints" for files.

An **md5sum** is a string of letters and numbers that acts like a fingerprint for a file. If two files have the same MD5 sum, the files are exactly alike - which is why MD5 "fingerprints" can verify whether or not your downloaded file got corrupted in transit.

```
root@kali:~/Desktop# md5sum secretrendezvous.docx
9e423e11db88f01bbff81172839e1923  secretrendezvous.docx
```

sha1sum is a computer program that calculates and verifies SHA-1 hashes. It is commonly used to verify the integrity of files. It (or a variant) is installed by default in most Unix-like operating systems. Variants include shasum (which permits SHA-1 through SHA-512 hash functions to be selected manually) and sha224sum, sha256sum, sha384sum and sha512sum, which use a specific SHA-2 hash function.

```
root@kali:~/Desktop/pcap_files/forensic-pcaps# sha1sum
secretrendezvous.docx
d60200be05625e4e7dcf9b40b2b5b598e30081a8
secretrendezvous.docx
```

Application Whitelisting

The SCADA/ICS environment presents many characteristics that can be exploited and difficult to protect:

- The system never really changes over time
- The system can be quite old (legacy)
- The system does the same process over and over again
- The system operates in a narrow envelope of expected norms
- The system does not require much application maintenance
- The system users/operators should not have exceptional privileges over the applications.

Application whitelisting can help out where other security measures may not be appropriate (i.e. anti-virus). The whitelist is a simple list of

applications that have been granted permission to run by an administrator. By default operating systems allow users to download and run any program. In application whitelisting, when an application tries to execute, it is automatically checked against the list and, if found, allowed to run. An integrity check measure, such as hashing, is generally added to ensure that the application is in fact the authorized program and not a malicious or otherwise inappropriate one with the same name. Application listing is similar to being part of a club; if you are not on the list, you don't get in.

Advantages

- Blocks most current malware.
- Prevents use of unauthorized applications.
- Does not require daily definition updates.
- Requires administrator installation and approval of new applications.

Disadvantages

- Requires performance overhead to enforce the whitelist (varies greatly depending on implementation).
- Requires regular maintenance of the whitelist to add new applications and remove ones that are no longer approved.
- Causes some users to be annoyed because they cannot download and run applications at will.

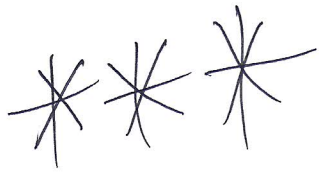
LO17: Describe the components of cybersecurity management.

Learning Objective 17

Security Controls

Over the years, many security standards and requirements frameworks have been developed n attempts to address risks to enterprise systems and the critical data in them. Unfortunately, most of these efforts have on compliance reporting diverting security program resources defense to paperwork.

The resources below offer controls and strategies, which may help you decide where to spend your money first to get the biggest return on investment.



- Center for Internet Security http://www.cisecurity.org/
  The Center for Internet Security (CIS) is a nonprofit organization focused on enhancing the cyber security readiness and response of public and private sector entities, with a commitment to excellence through collaboration. It provides actual system configuration commitment to excellence through collaboration and configuration recommendations (benchmarks) to increase the security posture of the individual system or device. Resources include secure configuration benchmarks, automated configuration assessment tools and content, security metrics and security software products certifications.
- SANS http://www.sans.org
  "Twenty Critical Controls for Effective Cyber Defense" http://www.sans.org/critical-security-controls/cag4-1.pdf

  "Top Cyber Security Risks" http://sans.org/top-cyber-security-risks/
- Australian Defense Signals Directorate (DSD.gov.au)
  "Top 35 Mitigation Strategies"
  http://www.dsd.gov.au/infosec/top35mitigationstrategies.htm

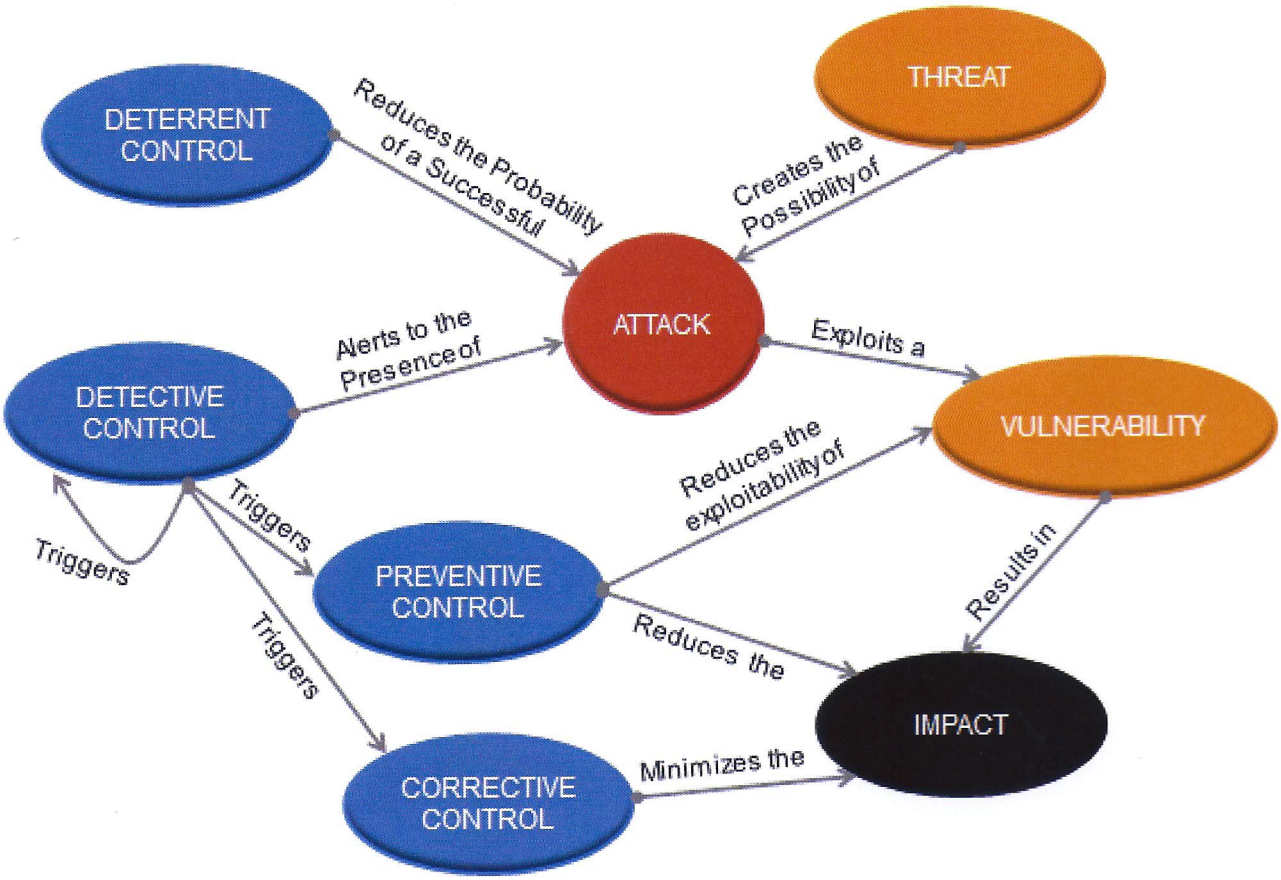
  [Note: As of April 2013, the Top 4 Strategies to Mitigate Targeted Cyber Intrusions are mandatory for Australian Government agencies. A guide for implementation for the top four: http://www.dsd.gov.au/infosec/top-mitigations/top-4-strategies-explained.htm.

Security Risk Management

Companies cannot eliminate all risks for two reasons. First, the threats that create risk are very dynamic. This includes internal and external threats. Second the investments required eventually result in diminishing returns. Instead of focusing time and resources on eliminating risk, a realistic goal should be to reduce risk via Risk

Management to a level that is acceptable to senior management and the board.

To reduce risk we must understand the relationship between vulnerabilities, threats, attack and impact. Risk can be seen as a function of vulnerabilities, threats and impact.



Mitigation Strategies

Strategies for Logging

It is increasingly likely that at some point you will have a security incident on your network. Take steps now to ensure you have the data you need to mitigate when it happens.

- Verify what your HMI is logging. If a set point is changed, would you know when, from where, and by whom?
- Baseline your network traffic. ICS traffic is very deterministic and perfect for network traffic flow baselining.
- Increase network monitoring and logging capabilities to allow the use of indicators to detect compromises and exfiltrated data.

- Ensure event logging (applications, events, login activities, security attributes, etc.) is turned on and monitored for identification of security issues.
- Secure logs, preferably in a centralized location, and protect them from modification.

Other Mitigations

- Network segmentation
- Protecting logon credentials for network hosts
- Audit network hosts for suspicious files
- Strict role-based access control
- Application whitelisting
- Preserve forensic data
- End user training (security awareness program)
- Work with your vendors to ensure equipement has required security controls.

Operations Security (OPSEC)

"OPSEC is simply denying an adversary information that could harm you or benefit them. OPSEC is a process, but it is also a mindset. By educating oneself on OPSEC risks and methodologies, protecting sensitive information becomes second nature.

OPSEC is unique as a discipline, because it is understood that the OPSEC manager must make certain decisions when implementing OPSEC measures. Most of these measures will involve a certain expenditure of resources, so an estimate must be made as to whether the assumed gain in secrecy is worth the cost in those resources. If the decision is made not to implement a measure, then the organization assumes a certain risk. This is why both OPSEC managers and leaders at all levels must be educated on and aware of the OPSEC process.

OPSEC is not only for Military or Government entities. More individuals and corporations are realizing the importance of protecting trade secrets, personal security, and intentions. Whatever the organization and purpose, OPSEC can, and will, increase the overall security posture."

- The Operations Security Professionals Association (http://www.opsecprofessionals.org/what.html)

The Three Rules of OPSEC

1. If you don't know the threat, how do you know what to protect?
2. If you don't know what to protect, how do you know you are protecting it?

SANS

SANS.org: Top Cyber Security Risks - Vulnerability Exploitation Trends http://www.sans.org/top-cyber-security-risks/trends.php

Homeland Security

3. If you are not protecting it (the critical/sensitive information), the adversary wins!

What are you telling the world?

The pieces of the puzzle can come from any or all of the six groups of sources shown below. Some of these require physical observation, while others can be done virtually over the web.

When we first started our training in 2004, one thing we saw is the unintentional leakage of information that a hacker community could use to fine tune their attacks. We saw several webpage's that contained system diagrams with details down to the brand and model number of ICS equipment deployed in asset owner's facilities. In some cases, there were system drawings containing actual IP addresses. Many of these available drawings were from organizations or companies that you would never expect to have this type of information on the web.

Supply Chain Considerations

Security principles should be considered when designing and procuring control systems products and services (software, systems, maintenance, and networks). This information should be a part of the procurement language used when ordering equipment, software or services. This does not forego the use of engineering practices. The system's prime requirements, functions, design, and expected behaviors need to be taken into account prior to adding or requesting security requirements.

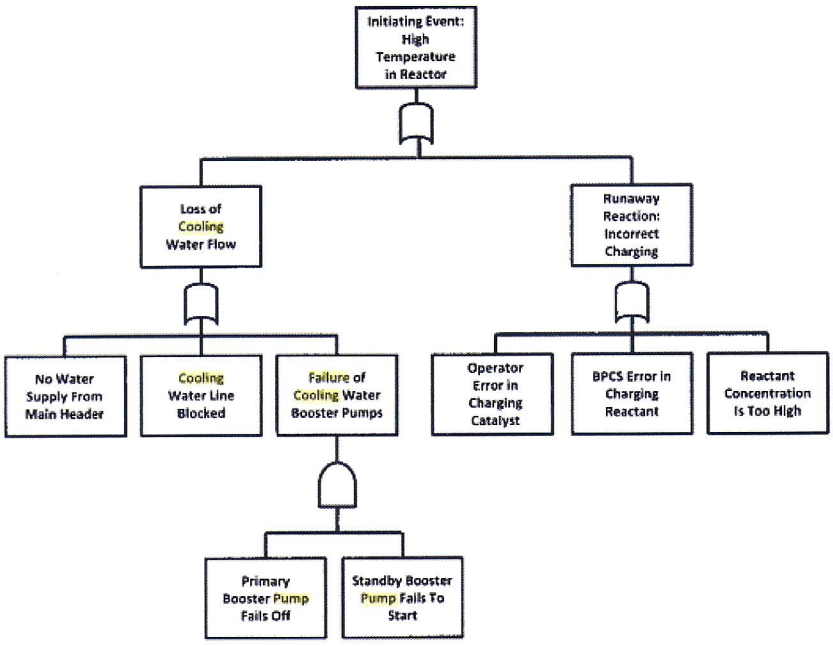
The purchaser is encouraged to work with the vendor(s) to identify risk mitigation strategies specific to their system that may include solutions outside of recommended practices. Many vendors are considered industry experts and are a valuable resource to the purchaser.

The U.S. Department of Homeland Security has created a document with information on security procurement language for control systems. This can be found at: https://ics-cert.us-cert.gov/sites/default/files/documents/Procurement_Language_Rev4_100809.pdf

Fault Trees and Attack Trees

In many engineering disciplines, fault tree analysis (**FTA**) is used to determine failure states, rates, costs, and risks. The fault tree helps break down the failure into simpler, easier to analyze steps. Sources of top level events include: problem/known error records, service outage analysis, potential failures from brainstorming, and what-if scenarios based on service level agreements. Below is a fault tree for a high temperature event in a nuclear reactor.

Fault Tree Analysis



Attack trees are the virtual world's version of fault trees. They provide a visual representation of possible attacks scenarios against a given target. This type of attack modeling is thought to have started in the

https://www.schneier.com/paper-attacktrees-ddj-ft.html

Bruce Schneier, **Secrets & Lies: Digital Security in a Networked World,** John Wiley & Sons, January 2004, ISBN: 0-471-45380-2

intelligence community in the late 1980's. The root of the tree (top node) is the goal of the attacker: deface systems, infect process and/or systems, denial of service, etc. The internal nodes make up the leaves of the tree and denote the sub goals the attacker must accomplish in order to reach the end goal. The tree grows as the additional sub goals the attacker must accomplish are identified. The addition of the sub goals is repeated as many times as necessary to achieve the level of detail and complexity necessary to analyze the attack scenario.

The idea was made popular in 1999 by Bruce Schneier in his paper "Attack trees: Modeling security threats." Mr. Schneier also wrote about attack trees in his book "Secrets & Lies: Digital Security in a Networked World."

The goal of attack tree analysis is to understand the various methods and paths that lead to a computer system being compromised. When you do the analysis of the attack you can estimate a 'cost for the attacker' by assigning numbers that could represent time, money, etc. The analysis tools roll these numbers up the tree towards the root allowing you to identify the least expensive attack path.

The following three conditions must be present for an attack to be successful:

- There must be vulnerabilities in the system being defended.
- The threat actor must have sufficient resources or capabilities to carry out the attack.
- The attacker must believe that if they succeed, there will be benefit (motivation factor).

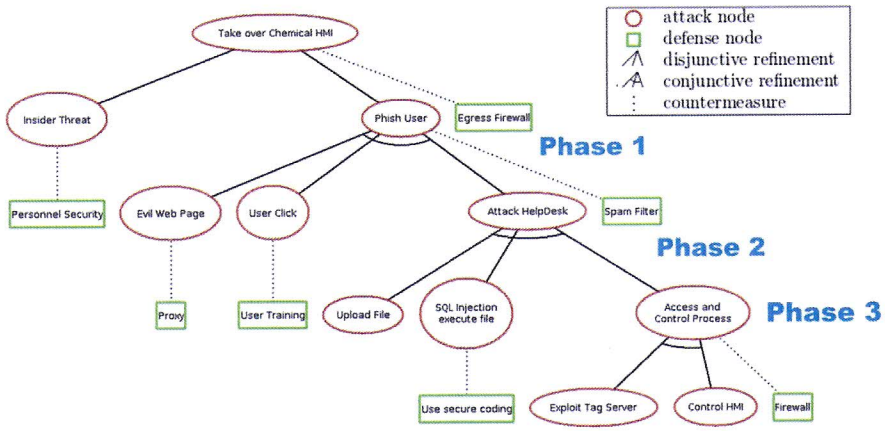
If the attacker has the resources to carry out ***ALL*** of the steps for a particular attack, the success of the attack is **possible**. If there is also the motivation to carry out the give attack *scenario*, then the attack *scenario* is **probable**. Those attack scenarios that require resources greater than envisioned for the adversary, are least probable and can be pruned from the attack tree.

Attack trees are built based on the view from the **attacker**. This is why you have to 'Turn the EVIL bit on'! You should not approach the model process based on defending the system.

Developing an attack tree can be a time consuming effort and just like a network drawing, it has a relative short lifetime. If the attack tree is not kept up to date with the emerging threats and defense techniques, its usefulness will diminish over time.

The following attack tree diagrams are for the INL Demo shown earlier this week.

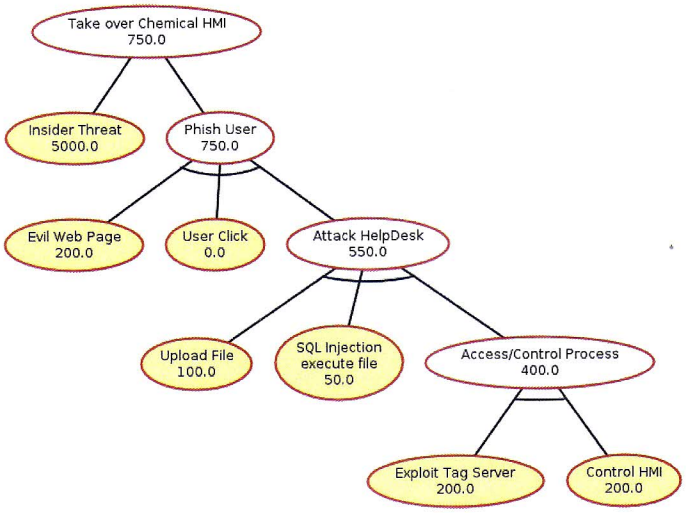
Attack Tree Analysis – INL Demo



Be aware, that the countermeasures are **assumed** to be correctly installed and configured.

Cost without Defense

Now that you have the attack scenario diagramed, it is time to come to grips with identifying the possible attacks and also the most probable attacks. In order to do that you have to assign some numbers to the sub goals an attacker must accomplish to be successful. These numbers could represent actual cost to accomplish, time, etc. Most of the attack tree analysis tools will compute overall costs for the attack sub goals allowing you to identify the cheapest attacks. These are the sub goals that should be fixed first.



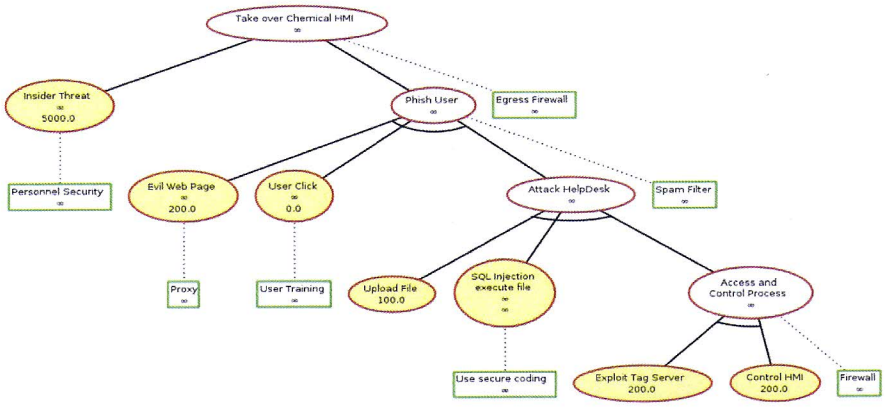
With the analysis tools, you can also play a lot of what if games by changing the numbers or adding/ removing counter-measures to get a probability range of various attack scenarios.

Below is the attack tree diagram showing the various costs for the attacks, without considering any countermeasures deployed. The bubbles in yellow are the sub goals you can assign costs to. The white bubbles show the rolled up costs from the sub goals below.

Homeland Security

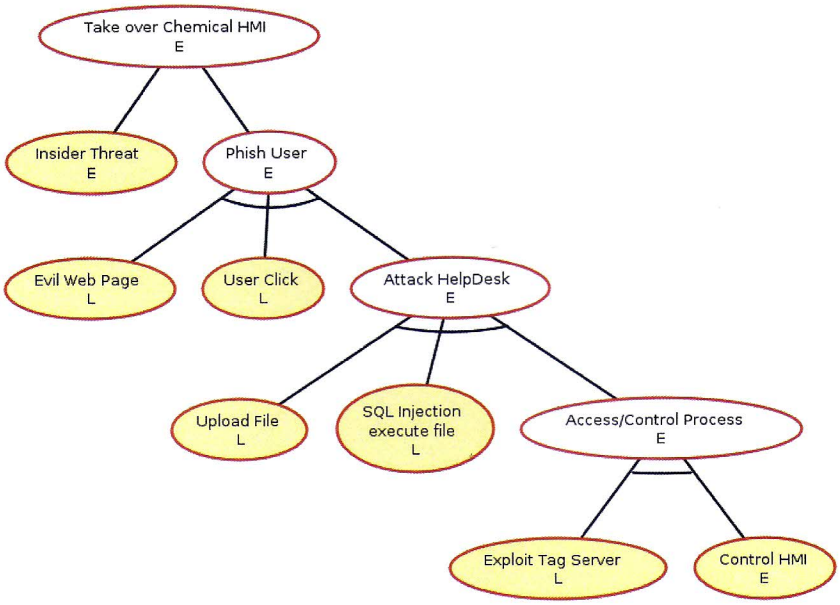
Cost with Defense

The next diagram shows the costs with the countermeasures considered.



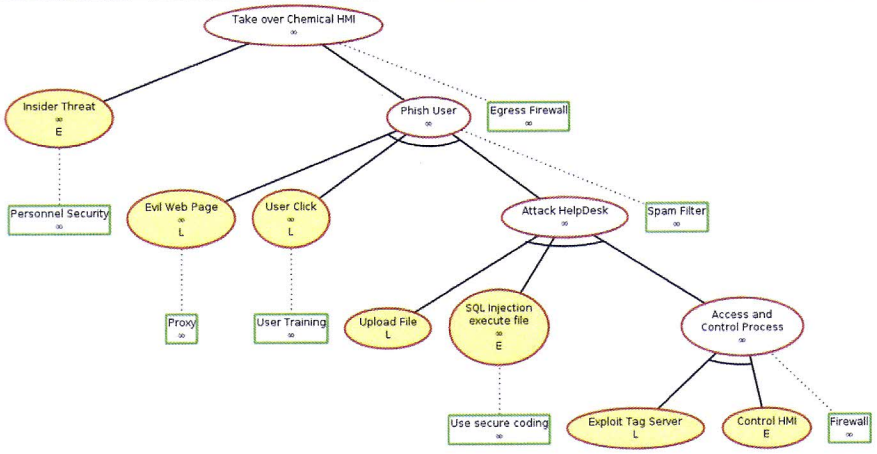
Attackers Expertise – No Defense

We now turn to evaluating the attackers expertise required for success.The following Attack Tree shows the attackers expertise level (L=low, M=Medium, H=High) as the metric. Again, in the first analysis, the countermeasures were not taken into account.



Attackers Expertise with Defense

In the next diagram, we account for the countermeasure begin deployed. Note the change in the expertise required.



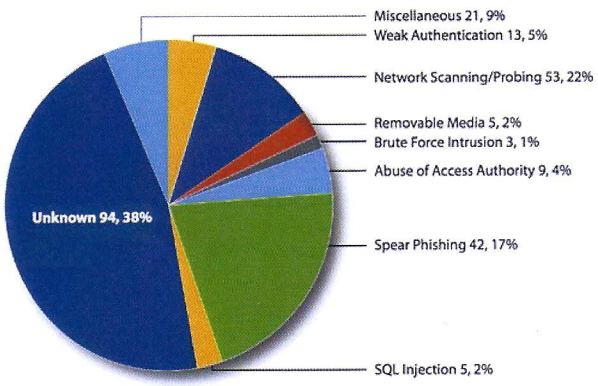
Incident Response

"The majority of incidents were categorized as having an "unknown" access vector. In these instances, the organization was confirmed to be compromised; however, forensic evidence did not point to a method used for intrusion because of a lack of detection and monitoring capabilities within the compromised network."

--ICS-CERT Monitor, Sep 2014 – Feb 2015.

A complete discussion on incident response would be a whole class in itself. However, a short discussion is warranted to point out some important aspects of incident response.

It can happen to you no matter what you believe your security to be. The graph below shows the types of intrusions/compromises for Sep 2014 – Feb 2015. The graph below shows the various forms of attack methods uncovered by the ICS-CERT, although worryingly the vast majority of attacks were untraceable. This is just the reports received by ICS-CERT during this time period. Many attacks go unreported or unnoticed.



Homeland Security

Incident response has two basic components. First, the actions you must take NOW to get things under control. Then those things you need to do afterwards to diagnose what happened and to clean up the mess. During an incident is not the time to develop a response plan. It is also not the time to gather a haphazard team together. The important word in the last sentence is team. Knowing the available skills and focus areas is critical when you have to parse out activities and assignments.

You need to have a solid plan with:

- Participants from all aspects of your business.
- Secure and alternate methods of communication.
- Scribe or scribes for each group within the team.
- A securable room where you can keep ACCURATE and COMPLETE information.
- Access to ALL of the logs and data.
- Known, certified clean computer systems.
- A person with the authority to unplug from the Internet.
- A practiced plan.
- Use a checklist for a starting point.

The scribe is probably the most important position. They keep a detailed log of what you have done, what assignments have been made, and what needs to be done. This will come in handy when you have to write the final incident report. The log also helps to keep track of the progress being made during the response events. Depending on the size of your team, you may want an overall scribe and a scribe for each of the teams.

Cybersecurity Incident First Aid

If you have ever taken a first aid course, you will remember that there is an order to treat the issues a victim might have. We will administer first aid to our virtual environment using the same methodology.

It is a good start to call in additional help like the **ICSCERT** field teams. This is our equivalent of calling the paramedics. It may take them a *few days to reach your location...time you can't afford to waste watching your information go out the* front door. Start collecting and preserving ALL of your log information and write it to DVDs. Make a copy for the authorities.

Bleeding - Information Leakage Out the Front Door

Many folks, those that do not understand (C-level) what is going on, think the answer to an attack is to fire back at what they think is the

attacker (actually the zombie).

There are three problems with this solution:

1. It's illegal! Because you are now attacking an innocent (though compromised) bystander.
2. It won't accomplish what you think it should. If you take out one zombie, there are many more to take its place. It could be Aunt Josie's or your grandmother's system you attack.

If there are enough zombies attacking you, you could end up inflicting a denial of service attack on your point of presence (Internet connection) to the Internet.

Instead, find the callback address in your logs. (Hint: Use your IDS rules.)

Breathing - Take a Breath and Evaluate Your Situation

You are now ready to move the victim to triage. During this step, take the time to do a thorough inventory of your environment. It is important that you do this very methodically. You do **NOT** want to leave a victim lying out in the field. If you do, the attack could just start up again. Take the time to do a thorough analysis of your logs and network traffic.

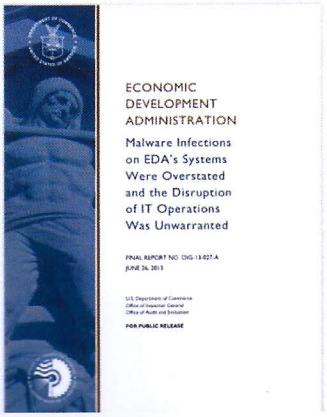
Poisoning - Now is the Time to Do the Hard Part - the Cleanup

This part might take expert help to actually clean up the systems that were involved and any physical damage that was done to your equipment.

- For the systems that have been compromised, save them for analysis by the ICS-CERT fly-away team. They can look for various artifacts (files, logs, changes, etc.) that can help in the overall cleanup and possibly shed some light on the attackers and their motive(s).
- When you reload your ICS systems use reputable sources, backups that have been written to media that cannot be changed; like CD/DVD copies (often referred to as the Gold Standard or Gold Disk).

Incident Response gone WRONG!

There is a report in the Documents folder of your Kali disc (OIG-13-027A.pdf) entitled: "ECONOMIC DEVELOPMENT ADMINISTRATION Malware Infections on EDA's Systems Were Overstated and the Disruption of IT Operations Was Unwarranted," FINAL REPORT NO. OIG-13-027-A, JUNE 26, 2013.



From the cover letter:

"We found (I) EDA based its critical incident response decisions on inaccurate information, (2) deficiencies in the Department's incident

response program impeded EDA's incident response, and (3) misdirected planning efforts hindered EDA's IT system recovery."

On December 6, 2011, the US-CERT notified the EDA and NOAA that they had detected some malware on systems belonging to the two agencies. NOAA remediated the malware and placed the effected systems back into service on Jan 12, 2012.

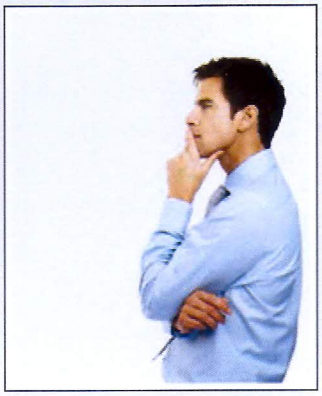
On January 24, 2012, EDA, believing it had a "widespread malware infection" requested their systems be isolated for the Department of Commerce network. This shut down communications for the ECA.

It turns out there was no evidence of malware on the systems. There were several communication problems between organizations. There were many cases of not verifying the data. Despite only finding common malware infections, EDA's management and CIO remained convinced that there could be extremely persistent malware somewhere in EDA's IT systems.

The bottom line is that $2.7 million was spent on the event, including $175,000 in new equipment since the old equipment was destroyed because of a minor piece of malware.

This report is a good read for all incident response teams. It contains lessons learned from an incident response that didn't go well.

Review of Session 6



- Defense is difficult.
- Analyze **ALL** the applications and services on your network.
- Use intrusion detection.
- Review and modify your network architecture – document.
- Although painful, someone has to review ***all*** the logs.
- If possible, upgrade to modern hardware and software.
- You have to be RIGHT 100% of the time…No Slacking!
- **Question:** What is our goal in securing these systems?

Attacker Tactics

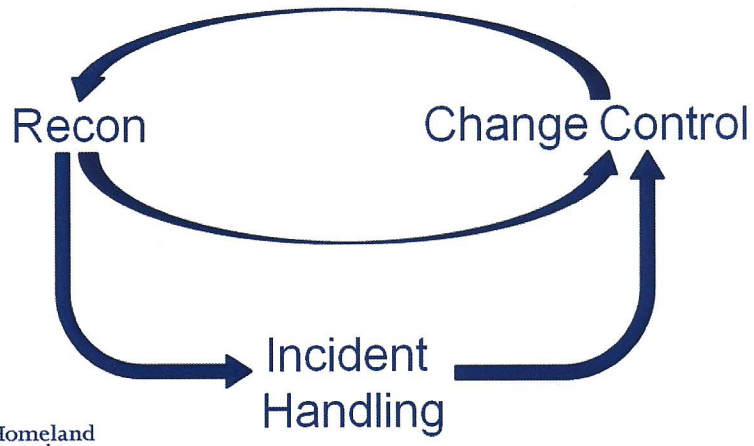
1. Recon
   - Learn all you can about the target
2. Find vulnerability
   - Social engineering / Email phishing / vulnerability scanning / logging in / etc.
3. Gain direct access - exploit vulnerability
   - Carry out goal
     - Privilege escalation / install trojan / copy files / delete files / cause damage
4. Cover tracks and install backdoor
5. Repeat

Homeland Security

3

Homeland Security

Defender Tactics



Homeland Security

4

Defender Tactics

Recon

- Know your network
  - Topology
  - Data paths
  - Patch levels
  - OS/Applications
  - Passive Discovery
  - Active Discovery
  - Vulnerability Scans
- Know your public face
  - Website
  - FTP access
  - Data egress

Change Control

- Patch levels
- Firewall rules
- Network topology
- Config files
- Other tactics?
  - Think outside the box
  - Work with vendors
    - User group meetings

Homeland Security

5

Defender Tactics

Incident Handling

- Identify (intruder / virus / worm / etc.)
- Verify (i.e., show proof of malicious intent)
  - tcpdump / wireshark
  - Files created and/or modified
  - Firewall logs
  - IDS logs
  - Active accounts & privileges
- Take action
  - tcpView, CurrPorts, etc.

Homeland Security

6

Homeland Security